Best Practices for Rich Media Ads in Asynchronous Ad Environments

Released October 2008

These Best Practices have been developed by the IAB Rich Media & AJAX Working Group with guidance from the IAB Ad Operations Council.

About the IAB Rich Media & AJAX Working Group:

The IAB Rich Media & AJAX Working Group consist of key IAB member companies working together in order to create best practices aimed at standardizing the interface between rich media ads and sites that load such ads in an Asynchronous Javascript and XML (AJAX) or other type of dynamic environment.

Key Contributors:

AOL
CNET Networks / CBS Interactive
Google
Microsoft
Yahoo!, Inc.

About the IAB Ad Operations Council:

The Ad Ops Council is dedicated to improving the operational efficiency of interactive advertising. Ad Ops Council working groups regularly include agency-side representatives to help improve communication, understanding, and work process in many areas of the buyer-seller relationship.

A full list of Council member companies can be found at:
http://www.iab.net/member_center/35088?iabid=a0350000002Cmy1AAC

This document can be found on the IAB website at:

www.iab.net/ajaxrichmedia

IAB Contact Information:

Jeremy Fain
Vice President of Industry Services, IAB
212-380-4724
Jeremy@iab.net

Executive Summary

The rise of rich internet applications and environments, programmed in frameworks such as Asynchronous Javascript and XML (AJAX), has enabled advertising to be delivered dynamically into a webpage without a full page reload. While dynamic ad loading works well with standard ads using simple animations, rich media advertisements that can expand, hover, and exist outside of a standard ad frame are more problematic because of their use of more complicated Javascript and Dynamic HTML (DHTML) programming.

These Best Practices attempt to establish a standard set of Rich Media implementation rules for rich media ad vendors, creative development teams, and publishers. In the current non-standardized publisher environments, Rich Media ads must either be built uniquely to each publisher's rich internet environments or are not accepted at all. This is because, when served, the Rich Media ads may not render correctly or additional, dynamically loaded Rich Media ads may render incorrectly because of prior loaded ads.

Adhering to these Best Practices will enable:

- Rich Media vendors to implement standard elements within their development and publishing tools that will automatically insert coding accepted and rendered correctly by compliant publishers
- Creative Development teams to develop their own Rich Media that will also be accepted and rendered correctly by compliant publishers
- Publishers to develop a standard environment that will accept and correctly render all Rich Media ads designed to compliant specifications

Although this document is highly technical and is intended for technical resources at these three types of companies, it is important for everyone to understand the implications of these Best Practices. Rich Internet Applications and other rich environments are enabling richer user experiences throughout the Internet. Adoption of these Best Practices will allow more publishers to improve their content and services for their users by enabling the serving of the widest range of advertising products.

Best Practices

Rich Media ads are typically served with Javascript ad calls that do not work with AJAX and other dynamic coding frameworks. In addition, the ads may make calls to document.write which interferes with the rest of an AJAX-based webpage. In order to solve these problems, the Ad Ops Council recommends a **Friendly IFrame (FIF)** solution which is further detailed below:

Publisher Creation of Friendly IFrame

There are two methods that Publishers can use in their environments to create the Friendly IFrame, based on operating systems and browsers:

- **For Windows Operating Systems with Internet Explorer 6.0 or Higher**
  - Create an IFrame element and place it into the document with SRC set to about:self
  - Create a Script element and place it into the IFrame document with SRC set to the ad delivery system
  - Generate a variable called inDapIF = true in the IFrame document to identify to the ad code that it is running inside the Dynamic IFrame.

- **Windows Operating Systems with Internet Explorer 6.0 or Higher OR Firefox 2.0 or Higher**
  - Create an IFrame element and place it into the document with SRC set to a small adpage file in the same domain as the page.
  - Create a Script element and place it into the IFrame document with SRC set to the ad delivery system
  - Generate a variable called inDapIF = true in the IFrame document to identify to the ad code that it is running inside the Dynamic IFrame.

NOTE: The IFrame should not be created using document.write as this may cause problems with AJAX pages. The Script elements can be created using document.write as they are within the IFRAME and will not interfere with the page content.

The first method has the advantage of not requiring a hosted file, but the second method has been found to be more reliable and it supports the increasingly common Firefox browser.

<u>If the browser or operating system of the user is not one of those listed above:</u>

- Use a normal JavaScript based ad call which will not support reloading
- Use an image based ad call that will support reloading, but will not support rich media
- Use a standard IFrame call, which can reload with in-banner rich media only but does not provide support for expandables

Rich Media Vendor / Creative Team Support of Publisher Friendly IFrames

In order for ads to work correctly within these best-practice environments, the following practices should be followed by the ad development team:

- The expanding ad *should* remain in the DIV that it is loaded in. This is even when the ad expands outside of the IFrame. It allows the AJAX ad calls to clear the ad by clearing the DIV

- If the expanding ad must create code outside of the DIV, such as adding itself to the BODY tag, then it must monitor the DIV it should have stayed in. When the DIV is cleared, it should automatically eliminate any code it created outside of the DIV.
- As mentioned earlier, the ads must have no "document.write" elements except inside the IFrame itself.

Additional Technical Details for FIF Solution

Cross-site Scripting Limitations	No (vendor ad code must support the standard FIF)
Expanding Limitations	No (vendor ad code must support the standard FIF)
Dynamic Ad Size Placement Support	Yes
Ad Load Asynchronous to Page Load	Yes
Publisher Side File (PSF) Required	No (unless vendor solution is to use a PSF in this scenario)
IAB Compliant Impression Tracking	Yes
Browser Support for Asynchronous Loading	IE 6.0 and higher, Firefox 2.0 and higher
Variable details	inDapIF = true

An example of a page where a Friendly IFRAME implementation can be viewed and various vendors AJAX tags can be tested:
http://browsertest.web.aol.com/ads/tstRMIF.htm